REMARKS

Applicant's remarks, below, are preceded by quotations of related comments of the examiner, in small, bold-face, type.

The drawings are objected to because Figures 1 ...? are not designated by a legend such as "Prior Art". The legend is necessary in order to clarify what applicant's invention is. MPEP 608.02(g). Appropriate correction is required.

A copy of figure 1, redmarked with a Prior Art legend, is attached.

Objections to the claims 1-25:
"Knowbot" is a registered trademark subject to change over time (see page 4 of instant application). Applicant has the burden of distinctly claiming the instant invention. Appropriate correction is required.

Amendments have been made to make clear that the claims refer to any kind of mobile program, not only so-called knowbots.

Claims 1-19 are rejected under 35 U.S.C, 103 as being unpatentable over Antes, Gary M., "Let your 'knowbots' do the walking", Computerworld, May 13, 1991, pp(2), in view of Steinberg, Don, "Demon knowbots (intelligent software robots)," PC-Computing, v3, n1, pp(4), Jan 1990.

As per claim 1: Antes discloses the invention substantially as claimed:

Antes teaches a method for use in a distributed system for processing a knowbot program that has the ability to move from node to node in the distributed system [e.g., page 1, line 24].

Antes teaches in an operating environment in each of the nodes, providing service facilities useful to the knowbot program [e.g., page 1, line 30].

However, Antes does not *explicitly* disclose the following additional limitations:

Steinberg teaches in the operating environment running a supervisor process [e.g., administrative knowbots, page 3, line 3] that enables the knowbot program to make use of the service facilities but does not permit access by the knowbot program to facilities of the operating environment [page 3].

It would have been obvious to one of the ordinary skill in the art at the time the invention was made to improve upon the system taught by Antes by implementing the improvements detailed above because it would provide

Antes's system with the enhanced capability of keeping unauthorized users out [e.g., page 3, line 4].

Applicant has amended claim 1 to make clear that the supervisor, running in the operating environment in each node, provides the mobile program with the capability of indirect access to the service facilities in the operating environment.

Antes and Steinberg each comment broadly, without specifics, on certain aspects of applicant's technical work based on applicant's 1988 report.

Applicant believes that the examiner has read too much into the brief cited passage in Steinberg. What Steinberg says is that "administrative Knowbots...would police the system, keeping unauthorized users out..." (emphasis added), thus broadly implying some control over the use of the system. But Steinberg does not disclose or suggest that the administrative Knowbots, like the supervisor process of claim 1, would allow the mobile program indirect access to make use of the service facilities. Thus, even if Steinberg were combined with Antes, the result would not be the combination of claim 1 or any suggestion of that combination.

As per claim 2: Steinberg teaches creating a bastion object in the unrestricted environment to protect the unrestricted environment and passing it into a restricted environment within which the knowbot program is running [e.g., administrative knowbots, page 3, line 3].

As per claim 3: Steinberg teaches the bastion object provides an interface for the knowbot program to access the service facilities in a safe manner and which is substantially the same interface as the interface that the service facilities provide in the unrestricted environment [e.g., administrative knowbots, page 3, line 3].

As per claim 4: "Official Notice" is taken that the use of type checking on all methods calls made by a program to a service facility is well known in the art (e.g., Java run-time byte code type checking) [M.P.E.P. 2144.03].

Claims 2, 3, and 4 are patentable for at least the same reasons as claim 1.

As per claim 5: This claim is rejected for the same reasons detailed above in the rejection of independent claim 1, and also for the following additional reasons:

Applicant disagrees that the same reasons apply to claim 5.

> **Antes teaches a method for use in a distributed system for processing a knowbot program that executes in one node of the distributed system, may be interrupted at almost any point in its execution, and may be moved to another node of the distributed system for further execution [e.g., page 1, line 24].**

Applicant does not see where the cited part of Antes discloses or suggests interrupting a mobile program at almost any point in the execution, and conducting further execution at another node. What the quoted passage says is only that some mobile programs travel about and may transport other mobile programs.

> **Antes teaches in the one node, capturing state and program code of the knowbot program to the other node and continuing execution at the other node from the point of interruption based on the captured state and the program code [e.g., page 1, lines 24, 31, 40-44].**

Again applicant finds that the cited lines include no disclosure or even suggestion of capturing a current state, achieving the state to another node, or continuing execution at the other node based on the captured state.

> **As per claim 6: Antes teaches also delivering with the captured state and the program code a transported file system or other information created during execution of the knowbot program [e.g., page 1, lines 40-44].**

Claim 6 focuses on the delivery of information in a transported file system or other information useful for continued execution of the mobile program. Antes neither discloses nor suggests such a feature. Antes merely mentions that results of a query are handed off without identifying a mechanism for doing so.

> **As per claim 7: Antes inherently teaches the information in the transported files system or other information is accessible without executing the knowbot program [e.g., the databases disclosed by Antes are also manually accessible by the user**

without the use of knowbots; knowbots merely automate the searching task, pages 1-2].

As per claim 8: "Official Notice" is taken that the use of an encoding scheme of a language interpreter is notoriously well known in the art (e.g., the Java virtual machine run-time byte code interpreter) [M.P.E.P. 2144.03].

Claims 7 and 8 are patentable for at least the same reasons as the claims on which they depend.

As per claim 9: This claim is rejected for the same reasons detailed above in the rejection of the preceding independent claims, and also for the following additional reasons:

Applicant disagrees that the same reasons apply to claim 9.

Antes teaches a method for enabling communication with a knowbot program running in a distributed system, a knowbot service station [e.g., a knowbot policeman as disclosed by Antes, page 1, line 39], an extension, or another application [e.g., page 1, lines 39-44].

Antes teaches providing a connector mechanism [e.g., communications knowbot as disclosed by Antes] which permits each of the knowbot programs, knowbot service stations, extensions, and other applications to identify services that it provides, and permits each of them to find services that it needs and enabling knowbot programs to communicate with knowbot service stations via connector objects [e.g., inherently associated with the communications knowbots disclosed by Antes to effect communications between knowbots] associated with the connector mechanism [page 1].

Applicant believes that the examiner is reading too much into what Antes says. Antes does not disclose or suggest a connector mechanism let alone a connector mechanism having the features recited in claim 9.

As per claim 10: The combination of Antes as modified by Steinberg teaches the connector object (as discussed above) is provided by a supervisor process [e.g., Administrative Knowbots as disclosed by Steinberg, page 1, line 22] running in the distributed environment and the connector object prevents uncontrolled access to a needed service [e.g., Steinberg, page 3, line 4].

As per claim 11: Antes inherently teaches the connector mechanism which effects the functions of a connector broker and connector manager [e.g., see the "communications knowbot" as disclosed by Antes, page 1, line 41].

As per claim 12: "Official Notice" is taken that the use of objects consisting of data types (and methods) is notoriously well known in the art [M.P.E.P. 2114.03]. A typed object is not a patentably distinct limitation, per se, as all objects are associated with a particular type or class.

Claims 10, 11, and 12 are patentable for at least the same reasons as the claims on which they depend.

As per claim 13: This claim is rejected for the same reasons detailed above in the rejection of the preceding independent claims, and also for the following additional reasons:

Applicant disagrees that the same reasons apply to claim 13.

Antes, as modified by Steinberg, teaches a method for enabling negotiation between two unrelated knowbot programs, knowbot service stations, extensions, or other applications, in a distributed system, comprising:

in an operating environment in a node of the distributed system, receiving information from one of the two knowbot programs, knowbot service stations, extensions, or other applications, concerning a transaction offered to other knowbot programs, knowbot service stations, extensions or other knowbot programs, knowbot service stations, extensions or other applications [e.g., Antes, pages 1, lines 27-32, page 2, line 1],

If the examiner means that the clone provides information concerning a transaction offered, applicant disagrees that Antes either discloses or suggests a clone "providing information concerning a transaction offered".

in an operating environment in the node, receiving information from the other of the two knowbot programs, knowbot service stations, extensions, or other applications concerning a transaction in which the other of the knowbot programs, knowbot service stations, extensions, and other applications wishes to engage [e.g., see clone Knowbot, as disclosed by Antes, page 2, line 2],

- **notifying the other knowbot program, knowbot service station, extension, or other application of the one knowbot program, knowbot service station, extension, or other application [e.g., Antes, page 1, lines 41-44],**

- **enabling the two knowbot programs, knowbot services stations, extensions, or other applications to communicate concerning the transaction [e.g., see communications knowbot as disclosed by Antes, page 1, lines 41-45].**

As per claim 14: Antes teaches the information is received from the knowbot programs by a third knowbot program [e.g., page 1, line 30].

While Antes does mention the existence of three mobile programs, he does not disclose or suggest that those programs engage in the information passing that is recited in claim 13.

As per claim 15: This claim is rejected for the same reasons detailed above in the rejection of the preceding independent claims, and also for the following additional reasons:

Applicant disagrees that the same reasons apply to claim 15.

Antes, as modified by Steinberg, teaches a method for enabling action in an operating environment in a distributed system with respect to a knowbot program which is programmed in a language that is not fully supported by the operating environment, comprising:

- **labeling a knowbot program to identify operating environment features required for full support of the knowbot program [e.g, Antes, page 1, lines 41-44],**

Applicant disagrees that the cited lines disclose or suggest anything about identifying operating features required for full support of a mobile program. The examiner may have confused the claim 15 words "operating environment features" with Antes' "specific database".

- **in an operating environment, examining the labeling of the knowbot program to determine whether the operating environment supports all of the identified features, and taking an action based on whether all the identified features are supported [e.g., Antes, page 1, lines 41-44].**

Applicant submits that Antes does not identify a client as an operating environment, does not identify how a mobile program is labeled, and does not suggest examining labeling to determine whether the operating environment supports identified features.

As per claim 16: Antes teaches the action comprises sending the knowbot program to another operating environment for processing [e.g., page 1, lines 30-32].

As per claim 17: Antes teaches the action comprises retrieving non-program specific data from the knowbot program [e.g., page 1, lines 30-32].

Claims 16 and 17 are patentable for at least the same reasons as claim 15.

As per claim 18: This claim is rejected for the same reasons detailed above in the rejection of the preceding independent claims, and also for the following additional reasons:

Applicant disagrees that the same reasons apply to claim 18.

Antes, as modified by Steinberg, teaches a method for aiding communication with a knowbot program executing in operating environments provided at nodes of the distributed system (as discussed in the rejections detailed above), comprising

- **maintaining a name space that uniquely identifies types of information to be interchanged [e.g., Antes, page 1, line 41, i.e., a name is inherently associated with the specific format required, as taught by Antes],**

- **using a name within the name space to identify the type of information to be interchanged [e.g., Antes, page 1, line 41].**

As per claim 19: Antes teaches the knowbot program registers an interface which includes the name of a type of information that is to be interchanged [e.g., Antes, page 1, line 41, i.e., a name is inherently associated with the specific format required, as taught by Antes].

Applicant disagrees that Antes describes or suggests the use of name spaces or the registration of user interfaces.

Claims 20-25 are rejected under 35 U.S.C. 103 as being unpatentable over Antes, Gary M., "Let your 'knowbots' do the walking", Computerworld, May 13, 1991, pp(2), in view of Steinberg, Don, "Demon knowbots (intelligent software robots)," PC-Computing, v3, n1, pp(4), Jan, 1990, and further in view of Rasmus, Daniel W., "Intelligent Agents", PC AI, Jan/Feb 1995, pp(8).

As per claim 20: Antes & Steinberg disclose the invention substantially as claimed, as discussed above.

Applicant disagrees that Antes and Steinberg disclose the invention substantially as claimed.

However, Antes & Steinberg do not *explicitly* disclose the following additional limitations:

Rasmus teaches a method for controlling the timing of execution of an action associated with a knowbot program running in an operating environment provided at a node of a distributed system, comprising,

- **providing a trigger protocol in the operating environment [page 5, as marked, IBM IntelliAgent for Windows],**
- **enabling the knowbot program (i.e., agent program) to register a condition with the operating environment [inherent],**
- **causing the operating environment to trigger the execution of the action upon the occurrence of the condition [page 5, as marked].**

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve upon the system taught by Antes and Steinberg by implementing the improvements detailed above because it would provide the system taught by Antes & Steinberg with the enhanced capability of an event-driven interface agent that monitors directories, launches applications, manages files, and sorts, manages, and responds to email wherein the agent is triggered by information events and by time events [e.g., page 5, as marked].

As per claim 21: Rasmus inherently teaches the trigger protocol defines trigger statements each of which identifies at least the condition and the action [page 5, as marked, IBM IntelliAgent for Windows].

As per claim 22: Official Notice is taken that the use of a table of registered trigger expressions responsive to designated events is well-known [M.P.E.P. 2144.03].

As per claim 23: Rasmus teaches execution is triggered by a program contained in the knowbot (i.e., agent) program [page 5, as marked, IBM IntelliAgent for Windows].

Without conceding the examiner's position, applicant has cancelled claims 20 through 23.

As per claim 24: This claim is rejected for the same reasons detailed above in the rejection of the preceding independent claims, and also for the following additional reasons:

Applicant disagrees that the same reasons apply.

Antes & Steinberg & Rasmus teach a method for controlling interaction between a knowbot program and an application running in an operating environment provided at a node of a distributed system, as detailed in the rejections above. "Official Notice" is taken that the use of registered trusted programs is well known in the art [M.P.E.P. 2144.03].

Claim 24 does not cover every use of registered trusted programs. Claim 24 requires that the trusted portion be part of the operating environment and that this trusted portion provide trusted services to the mobile program. It also requires that what is required to be registered as trusted is portions of the application running in the operating environment. Only if those portions have been registered is indirect interaction via the operating environment permitted between the mobile program and the application. This combination of features is neither disclosed nor suggested in Antes, Steinberg, or Rasmus.

As per claim 25: This claim is rejected for the same reasons detailed above in the rejection of the preceding independent claims, and also for the following additional reasons:

Applicant disagrees that the same reasons apply.

"Official Notice" is taken that the use of safe extensions to an operating environment is well known in the art (e.g., the Java virtual machine run-time environment wherein unsafe Java program which access memory directly are preempted from doing so by Java virtual machine) [M.P.E.P. 2144.03]

While it may be true that notion of safe extensions to an operating system is well known today, applicant questions that it was well known as of the filing of this application and asks the examiner to provide clear evidence to support his position.

> **Objective-Reducing and Simplifying the areas of disagreement: The Examiner solicits Applicant's cooperation in *reducing and simplifying the areas of disagreement* by doing the following: 1) amending the independent claims in a manner fully supported by Applicant's specification to *clearly distinguish* over the prior art of record, AND/OR 2) directing *clear* and *concise* arguments to the *specific claim language and claim elements* that Applicant feels are not fairly taught or suggested by the cited prior art of record. Applicant should cancel claims where appropriate. Applicant should preferably avoid arguing general differences between the cited references and the instant invention as disclosed in the specification. Your cooperation is appreciated.**

> **Requested Format of Amended Claims: Please help expedite the prosecution of this application by including the text of all claims which remain in the case in your amendment. Please label each amended claims (AMENDED), or (TWICE AMENDED), or (THREE TIMES AMENDED), etc., after the claim number. Please label each unchanged claims (UNCHANGED) after the claim number [meaning the claim is the same as originally filed]. Please label each cancelled claim (CANCELLED) after the claim number. The text of a cancelled claim does not need to be included. This format is not mandatory, however, it will help expedite the processing of your application. Your cooperation is appreciated.**

Applicant has tried to comply with these sensible requests.

> **The Examiner requests that your amendment response be in paper form accompanied by a 3 1/2 inch IBM format floppy disk which contains a file copy of your amendment response in Adobe Acrobat PDF format (preferred), or in any version of Microsoft Word or WordPerfect, or in ASCII text format. Please include all pending claims, as detailed above. Only the paper copy will be entered--your floppy disk file will be considered a duplicate copy. Signatures are not required on the disk copy. The floppy disk is not mandatory, however, your cooperation is appreciated.**

A diskette in Word 2000 format is enclosed.

The dependent claims are patentable for the same reasons as the independent claims on which they depend.

Applicant asks that all claims be allowed. Please apply any excess charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: 1/28/2

David L. Feigenbaum
Reg. No. 30,378

Fish & Richardson P.C.
225 Franklin Street
Boston, Massachusetts 02110-2804
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

Version with markings to show changes made

1. (Amended) A method for use in a distributed system for processing a [knowbot] mobile program that has the ability to move from node to node in the distributed system comprising

in an operating environment in each of the nodes, providing service facilities useful to the [knowbot] mobile program, and

in the operating environment, in each of the nodes, running a supervisor process that [enables] allows the [knowbot] mobile program indirect access to make use of the service facilities [but does not permit direct access by the knowbot program to facilities of the operating environment].

2. (Amended) The method of claim 1 further comprising creating a bastion object in [the] an unrestricted environment to protect the unrestricted environment and passing it into a restricted environment within which the [knowbot] mobile program is running.

3. (Amended) The method of claim 2 in which the bastion object provides an interface for the [knowbot] mobile program to access the service facilities in a safe manner and which is substantially the same interface as the interface that the service facilities provide in the unrestricted environment.

4. (Amended) The method of claim 2 in which the bastion object performs type checking on all method calls made by a [knowbot] mobile program to a service facility.

5. (Amended) A method for use in a distributed system for processing a [knowbot] mobile program that executes in one node of the distributed system, may be interrupted at almost any point in its execution, and may be moved to another node of the distributed system for further execution, comprising

in the one node, capturing a current state of the [knowbot] mobile program execution,

delivering the captured state and program code of the [knowbot] mobile program to the other node, and

continuing execution at the other node from the point of interruption based on the captured state and the program code.

6. (Amended) The method of claim 5 further comprising

also delivering with the captured state and the program code information in a transported file system or other information useful for continued [created during execution] of the [knowbot] mobile program.

7. (Amended) The method of claim 6 in which the information in the transported file system or other information is accessible without executing the [knowbot] mobile program.

8. (Unchanged) The method of claim 5 in which the step of capturing comprises using an encoding scheme of a language interpreter.

9. (Amended) A method for enabling communication with a [knowbot] mobile program running in a distributed system, a [knowbot] mobile program service station, an extension, or another application, comprising

providing a connector mechanism which permits each of the [knowbot programs, knowbot service stations, extensions, and other applications] mobile program, the mobile program service station, the extension, and the other application to identify services that it provides, and permits each of them to find services that it needs, and enabling [knowbot programs] the mobile program to communicate with [knowbot] mobile program service stations via connector objects associated with the connector mechanism.

10. (Amended) The method of claim 9 in which each of the connector [object] objects is provided by a supervisor process running in the distributed [environment] system and [the connector object] prevents uncontrolled access to a needed service.

11. (Unchanged) The method of claim 9 in which the connector mechanism includes a connector broker and connector manager.

12. (Unchanged) The method of claim 9 in which the connector objects are data typed.

13. (Amended) A method for enabling negotiation between two unrelated [knowbot] mobile programs, [knowbot] mobile service stations, extensions, or other applications, in a distributed system, comprising

in an operating environment in a node of the distributed system, receiving information from one of the two [knowbot] mobile programs, [knowbot] mobile program service stations, extensions, or other applications, concerning a transaction offered to other [knowbot] mobile programs, [knowbot] mobile program service stations, extensions, or other applications,

in the operating environment in the node, receiving information from the [other] second of the two [knowbot] mobile programs, [knowbot] mobile programs service stations, extensions, or other applications concerning a transaction in which the [other] second of the [knowbot] mobile programs, [knowbot] mobile program service stations, extensions, and other applications wishes to engage,

notifying the [other knowbot] second mobile program, [knowbot] mobile program service station, extension, or other application of the one [knowbot] mobile program, [knowbot] mobile program service station, extension, or other application, and

enabling the two [knowbot] mobile programs, [knowbot services] mobile program service stations, extensions, or other applications to communicate concerning the transaction.

14. (Amended) The method of claim 13 in which the information is received from [the] two [knowbot] mobile programs by a third [knowbot] mobile program.

15. (Amended) A method for enabling action by an operating environment in a distributed system with respect to a [knowbot] mobile program which is programmed in a language that is not fully supported by the operating environment, comprising

labeling a [knowbot] mobile program to identify operating environment features required for full support of the [knowbot] mobile program,

in an operating environment, examining the labeling of the [knowbot] mobile program to determine whether the operating environment supports all of the identified features, and

taking an action based on whether all the identified features are supported.

16. (Amended) The method of claim 15 wherein the action comprises sending the [knowbot] mobile program to another operating environment for processing.

17. (Amended) The method of claim 15 in which the action comprises retrieving non-program specific data from the [knowbot] mobile program.

18. (Unchanged) A method for aiding communication with a mobile program executing in operating environments provided at nodes of a distributed system, comprising

maintaining a name space that uniquely identifies types of information to be interchanged as part of the communication, and

using a name within the name space to identify a type of information to be interchanged.

19. (Unchanged) The method of claim 18 in which the mobile program registers an interface which includes the name of a type of information that is to be interchanged.

Cancel claims 20 through 23 without prejudice.

20. (Cancelled) A method for controlling the timing of execution of an action associated with a mobile program running in an operating environment provided at a node of a distributed system, comprising

providing a trigger protocol in the operating environment,

enabling the mobile program to register a condition with the operating environment,

causing the operating environment to trigger the execution of the action upon the occurrence of the condition.

21. (Cancelled) The method of claim 20 in which the trigger protocol defines trigger statements each of which identifies at least the condition and the action.

22. (Cancelled) The method of claim 20 in which the operating environment maintains a table of registered trigger expressions for all mobile programs that have registered conditions.

23. (Cancelled) The method of claim 20 in which the execution is triggered by a program contained in the mobile program.

24. (Amended) A method for controlling interaction between a [knowbot] mobile program and an application running in an operating environment provided at a node of a distributed system, comprising

defining a trusted portion of the operating environment which provides trusted services to the [knowbot] mobile program,

requiring portions of the application running in the operating environment to be registered as trusted, and

permitting indirect interaction via the operating environment between the [knowbot] mobile program and the application running in the operating environment only if the portions of the application required to be registered have been registered.

25. (Amended) A method for enabling a [knowbot] mobile program to carry out defined functions including otherwise unsafe functions, [thorugh] through the use of extensions comprising

coding safe extensions to an operating environment and to [the] an interpretive language under which the [knowbot] mobile program runs, and permitting the [knowbot] mobile program to carry out the defined functions by making use of the extensions.